Managing and Monitoring Jobs

Eric Wright
Technology Evangelist, Podcaster

@DiscoPosse   www.discoposse.com   www.discopossepodcast.com

Overview

- Explore the Nomad job specification in some practical examples
- Create and run Nomad jobs using the web UI and CLI
- Monitor job status and health using both the web UI and CLI
- Review some more advanced job spec examples

The Nomad Job Specification

Demo

- Review a sample job specification to understand common parameters
  - create a simple job
  - configure health checks
  - customize restart behavior
  - configure affinity
  - configure priority

Demo

- Create and manage a simple job using the Nomad UI
    - plan the allocation
    - run a new job
    - scale a running job
    - stream logs from the task
    - restart a running job
    - stop a running job

Demo

- Create and manage a simple job using the Nomad CLI
  - dry run
  - run a new job
  - scale a running job
  - stream logs from the task
  - restart a running job
  - stop a running job

Using Deployments

Deployments and Job Updates

Deployments enable controlled updates for Nomad jobs

Rolling updates

Blue/Green and canary

Exit signals

Deployment Configuration

redis.nomad

```
# Configure the job to do rolling updates
update {
    stagger = "10s"
    max_parallel     = 2
    min_healthy_time = "30s"
    healthy_deadline = "10m”
    canary = 1
}
```

Demo

- Create a deployment and manage job lifecycle during an application update

Summary

- Explored the Nomad job specification
- Created and ran Nomad jobs using the web UI and CLI
- Monitored job status and health using both the web UI and CLI
- Reviewed some more advanced job spec examples

Up Next:

Configuring Nomad for High Availability